

Durham Research Online

Deposited in DRO:

15 January 2016

Version of attached file:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Dantchev, Stefan and Martin, Barnaby (2014) 'Relativization makes contradictions harder for Resolution.', *Annals of pure and applied logic.*, 165 (3). pp. 837-857.

Further information on publisher's website:

<http://dx.doi.org/10.1016/j.apal.2013.10.009>

Publisher's copyright statement:

© 2013 This manuscript version is made available under the CC-BY-NC-ND 4.0 license
<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Additional information:

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full DRO policy](#) for further details.

Relativization makes contradictions harder for Resolution [☆]

Stefan Dantchev

*School of Engineering and Computing Sciences, Durham University,
Science Labs, South Road, Durham DH1 3LE, U.K.*

Barnaby Martin

*School of Science and Technology, Middlesex University,
The Burroughs, Hendon, London NW4 4BT, U.K.*

Abstract

We provide a number of simplified and improved separations between pairs of Resolution-with-bounded-conjunction refutation systems, $\text{Res}(d)$, as well as their tree-like versions, $\text{Res}^*(d)$. The contradictions we use are natural combinatorial principles: the *Least number principle*, LNP_n and an ordered variant thereof, the *Induction principle*, IP_n .

LNP_n is known to be easy for Resolution. We prove that its relativization is hard for Resolution, and more generally, the relativization of LNP_n iterated d times provides a separation between $\text{Res}(d)$ and $\text{Res}(d+1)$. We prove the same result for the iterated relativization of IP_n , where the tree-like variant $\text{Res}^*(d)$ is considered instead of $\text{Res}(d)$.

We go on to provide separations between the parameterized versions of $\text{Res}(1)$ and $\text{Res}(2)$. Here we are able again to use the relativization of the LNP_n , but the classical proof breaks down and we are forced to use an alternative. Finally, we separate the parameterized versions of $\text{Res}^*(1)$ and $\text{Res}^*(2)$. Here, the relativization of IP_n will not work as it is, and so we make a vectorizing amendment to it in order to address this shortcoming.

Keywords: Proof complexity, Lower bounds,
Resolution-with-bounded-conjunction, Parameterized proof complexity

1. Introduction

We study the power of relativization in *Propositional proof complexity*, i.e. we are interested in the following question: given a propositional proof system is there a first-order (FO) sentence which is easy but whose relativization is hard

[☆]Extended abstracts of results in this paper appeared as [9] at CSR 2006 and as [10] at CSR 2013. Several proofs, especially those omitted from [9], appear here for the first time.

(within the system)? The main motivation for studying relativization comes from a work of Krajíček, [19]. He defines a combinatorics of FO structure and a relation of covering between FO structures and propositional proof systems. The combinatorics contains all the sentences easy for the proof system. On the other hand, as defined in [19], it is closed under relativization. Thus the existence of a sentence, which is easy but whose relativization is hard, for the underlying proof system, would imply that it is impossible to capture the class of easy sentences by a combinatorics. Ideas of relativization have also appeared in [11, 2]. The proof, in fact refutation, system we consider is Resolution-with-bounded-conjunction, denoted $\text{Res}(d)$ and introduced by Krajíček in [18]. It is an extension of Resolution in which conjunctions of up to d literals are allowed instead of single literals. The tree-like version of $\text{Res}(d)$ is usually denoted $\text{Res}^*(d)$. Krajíček proved that tree-like Resolution, and even $\text{Res}^*(d)$, have combinatorics associated with it. This follows also from Riis's complexity gap theorem for tree-like Resolution [23], and shows that the sentences, easy for tree-like Resolution, remain easy after having been relativized.

The next natural system to look at is Resolution. It is stronger than $\text{Res}^*(d)$ for any d , $1 \leq d \leq n$ (equivalent to $\text{Res}^*(n)$, in fact, where n is the number of variables), and yet weak enough so that one could expect that it can easily prove some property of the whole universe, but cannot prove it for an arbitrary subset. As we show in the paper, this is indeed the case. The example is very natural, the *Least number principle*, LNP_n . The contradiction LNP_n asserts that a partial n -order has no minimal element. In the literature it enjoys a myriad of alternative names: the *Graph ordering principle* GOP, *Ordering principle* OP and *Minimal element principle* MEP. Where the order is total it is also known as TLNP and GT. It is not hard to see that LNP_n is easy for Resolution [7], and we prove that its relativization RLNP_n is hard. A more general result has been proven in [25]; however the lower bound there is weaker. We also consider iterated relativization, and show that the d th iteration $d\text{-RLNP}_n$ is hard for $\text{Res}(d)$, but easy for $\text{Res}(d+1)$. We go on to consider the relativization question for $\text{Res}^*(d)$, where the FO language is enriched with a built-in order. The complexity gap theorem does not hold in this setting [11], though we are able to show that relativization again makes some sentences harder. A variant of the *Induction Principle* gives the contradiction IP_n , saying that there is a property which: holds for the minimal element; if it holds for a particular element, there is a bigger one for which the property holds, too; and the property does not hold for the maximal element. We prove that the d th iteration of the relativization of the Induction principle, $d\text{-RIP}_n$, is easy for $\text{Res}^*(d+1)$, but hard for $\text{Res}^*(d)$. More precisely, our results are the following:

1. Any Resolution refutation of RLNP_n is of size $2^{\Omega(n)}$. Firstly, this answers positively to Krajíček's question. Secondly, observing that RLNP_n has an $O(n^3)$ -size refutation in $\text{Res}(2)$, we get an exponential separation between Resolution and $\text{Res}(2)$. A similar result was proved in [25] (see also [1] for a weaker, quasi-polynomial, separation). Our proof is quite simple compared with that of [25], where this separation is a corollary of a more

general result, and our lower bound is stronger.

2. $d\text{-RLNP}_n$ has an $O(dn^3)$ -size refutation in $\text{Res}(d+1)$, but requires $2^{\Omega(n^\epsilon)}$ -size refutation in $\text{Res}(d)$, where ϵ is a constant dependent on d . These separations were first proved in [25]. As a matter of fact, we use their method but our tautologies are more natural, and our proof is a little simpler.
3. $d\text{-RIP}_n$ has an $O(dn^2)$ -size $\text{Res}^*(d+1)$ refutation, but requires $\text{Res}^*(d)$ refutations of size $2^{\Omega(\frac{n}{d})}$. This holds for any d , $0 \leq d \leq n$. A similar result was proven in [15]. Again, our tautologies are more natural, while the proof is simpler.

The second part of the paper is in the area of *Parameterized proof complexity*, a program initiated in [12], which generally aims to gain evidence that $\text{W}[i]$ is different from FPT. Typically, i is so that the former is $\text{W}[2]$, though—in the journal version [13] of [12]—this has been $\text{W}[\text{SAT}]$ and—in the note [20]— $\text{W}[1]$ was entertained. In the $\text{W}[2]$ context, parameterized refutation systems aim at refuting *parameterized contradictions* which are pairs (\mathcal{F}, k) in which \mathcal{F} is a propositional CNF with no satisfying assignment of weight $\leq k$. Several parameterized (hereafter often abbreviated as “p-”) proof systems are discussed in [12, 3, 6]. The lower bounds in [12], [3] and [6] amount to proving that the systems p-tree-Resolution, p-Resolution and p-bounded-depth Frege, respectively, are not *fpt-bounded*. Indeed, this is witnessed by the *Pigeonhole principle*, and so holds even when one considers parameterized contradictions (\mathcal{F}, k) where \mathcal{F} is itself an actual contradiction. Such parameterized contradictions are termed “*strong*” in [6], in which the authors suggest these might be the only parameterized contradictions worth considering, as general lower bounds—even in p-bounded-depth Frege—are trivial (see [6]). We sympathize with this outlook, but remind that there are alternative parameterized refutation systems built from embedding (see [12, 13]) for which no good lower bounds are known even for general parameterized contradictions.

In the world of parameterized proof complexity, we already have lower bounds for p- $\text{Res}(d)$ (as we have for p-bounded-depth Frege), but we are still interested in separating levels p- $\text{Res}(d)$. We are again able to use the relativized Least number principle, RLNP_n to separate p- $\text{Res}(1)$ and p- $\text{Res}(2)$. Specifically, we prove that

4. (RLNP_n, k) admits a polynomial-sized in n refutation in $\text{Res}(2)$, but all p- $\text{Res}(1)$ refutations of (RLNP_n, k) are of size $\geq n\sqrt{(k-3)/16}$.

Although we use the same principle as in the first part of this paper, the classical proof of bottleneck counting does not adapt to the parameterized world, and instead we look for inspiration to the proof given in [6] for the *Pigeonhole principle*. For tree-like Resolution, the situation is more complicated. RIP_n , admits fpt-bounded proofs in $\text{Res}^*(1)$, indeed of size $O(k!)$, therefore we are forced to alter this principle. Thus we come up with the *relativized vectorized induction principle* RVIP_n . We are able to show that

5. (RVIP_n, k) admits $O(n^4)$ refutations in $\text{Res}^*(2)$, while every refutation in $\text{Res}^*(1)$ is of size $\geq n^{k/16}$.

Note that both of our parameterized contradictions are “strong”, in the sense of [6]. We go on to give extended versions of RVIP_n and explain how they separate $\text{p-Res}^*(d)$ from $\text{p-Res}^*(d+1)$, for $d > 1$.

This paper is organized as follows. We begin with the preliminaries. After this, we give the results about Resolution and $\text{Res}(d)$ complexity of the $d\text{-RNLP}_n$ (Items 1 and 2 above) in Section 3. We then give the results about $\text{Res}^*(d)$ complexity of the $d\text{-RIP}_n$ (Item 3) in Section 4. Moving to the parameterized world, we give our separation of $\text{p-Res}(1)$ from $\text{p-Res}(2)$ (Item 4) in Section 5, and our separations of $\text{p-Res}^*(d)$ from $\text{p-Res}^*(d+1)$ (Item 5) in Section 6. We then conclude with some final remarks and open questions.

2. Preliminaries

We use the notation $[n] := \{1, \dots, n\}$ and we denote by \top and \perp the Boolean values *true* and *false*, respectively. A *literal* is either a propositional variable or its negation. A d -conjunction (d -disjunction) is a conjunction (disjunction) of at most d literals. A *term* (d -term) is either a conjunction (d -conjunction) or a constant, \top or \perp . A $d\text{-DNF}$ is a disjunction of (an unbounded number) of d -conjunctions.

A $d\text{-CNF}$ is a conjunction of (an unbounded number) of d -disjunctions. Thus we may identify CNFs with sets of clauses. Since variables will often be written in Roman capitals, conjuncts, disjuncts, CNFs and DNFs will benefit from being written calligraphically.

As we are interested in translating FO sentences into sets of clauses, we assume that a finite n -element universe \mathbb{U} is given. The elements of \mathbb{U} are the first n positive natural numbers, i.e. $\mathbb{U} := [n]$. When we say “element” we always assume an element from the universe. We will not explain the translation itself; the details can be found in [24] or [18]. An example of this translation can be found at the beginning of Section 3.

2.1. Resolution and $\text{Res}(d)$

The system of Resolution aims to refute a set of clauses by inferring from them the empty clause (a logical contradiction). We will introduce this system through its generalization, due to Krajíček [18], to $\text{Res}(d)$.

$\text{Res}(d)$ is a system to refute a set of $d\text{-DNFs}$. There are four derivation rules. The \wedge -*introduction rule* allows one to derive from $\mathcal{P} \vee \bigwedge_{i \in I_1} \ell_i$ and $\mathcal{Q} \vee \bigwedge_{i \in I_2} \ell_i$, $\mathcal{P} \vee \mathcal{Q} \vee \bigwedge_{i \in I_1 \cup I_2} \ell_i$, provided $|I_1 \cup I_2| \leq d$ (\mathcal{P} and \mathcal{Q} are $d\text{-DNFs}$). The *cut* (or *resolution*) rule allows one to derive from $\mathcal{P} \vee \bigvee_{i \in I} \ell_i$ and $\mathcal{Q} \vee \bigwedge_{i \in I} \neg \ell_i$, $\mathcal{P} \vee \mathcal{Q}$. Finally, the two weakening rules allow the derivation of $\mathcal{P} \vee \bigwedge_{i \in I} \ell_i$ from \mathcal{P} , provided $|I| \leq d$, and $\mathcal{P} \vee \bigwedge_{i \in I_1} \ell_i$ from $\mathcal{P} \vee \bigwedge_{i \in I_1 \cup I_2} \ell_i$.

A $\text{Res}(d)$ refutation can be considered as a directed acyclic graph (DAG), whose sources are the initial clauses, called also axioms, and whose only sink is the empty clause. We will measure the size of a refutation as the number

of internal nodes of the graph, i.e. the number of applications of a derivation rule. Whenever we say “we refute an FO sentence in $\text{Res}(d)$ ”, we mean that we first translate the sentence into a set of clauses defined on a finite universe of size n , and then refute it with $\text{Res}(d)$. The size of the refutation is then a function in n . In the notation “ $\text{Res}(d)$ ”, d may be seen as a function in the number of propositional variables. Important special cases are $\text{Res}(\log)$ as well as $\text{Res}(\text{const})$. Clearly $\text{Res}(1)$ is (ordinary) Resolution. In this case, we have only usual clauses, i.e. disjunctions of literals. The cut rule becomes the usual resolution rule, and only the first weakening rule is meaningful. The letter d is reserved for use as relating to the parameter of some $\text{Res}(d)$.

2.2. $\text{Res}(d)$ as a branching program

If we turn a $\text{Res}(d)$ refutation of a given set of d -DNFs Σ upside-down, i.e. reverse the edges of the underlying graph and negate the d -DNFs on the vertices, we get a special kind of *restricted branching d -program*. We introduce them in generality (not just for $d = 1$) as we will use them in the part of the paper on parameterized proof complexity. The restrictions are as follows. Each vertex is labelled by a d -CNF which partially represents the information that can be obtained along any path from the source to the vertex (this is a *record* in the parlance of [22]). Obviously, the (only) source is labelled with the constant \top . There are two kinds of queries, which can be made by a vertex:

1. Querying a new d -disjunction, and branching on the answer: that is, from \mathcal{C} and the question $\bigvee_{i \in I} \ell_i$? we split on $\mathcal{C} \wedge \bigvee_{i \in I} \ell_i$ and $\mathcal{C} \wedge \bigwedge_{i \in I} \neg \ell_i$.
2. Querying a known d -disjunction, and splitting it according to the answer: that is, from $\mathcal{C} \wedge \bigvee_{i \in I_1 \cup I_2} \ell_i$ and the question $\bigvee_{i \in I_1} \ell_i$? we split on $\mathcal{C} \wedge \bigvee_{i \in I_1} \ell_i$ and $\mathcal{C} \wedge \bigvee_{i \in I_2} \ell_i$.

There are two ways of forgetting information. From $\mathcal{C}_1 \wedge \mathcal{C}_2$ we can move to \mathcal{C}_1 . And from $\mathcal{C} \wedge \bigvee_{i \in I_1} \ell_i$ we can move to $\mathcal{C} \wedge \bigvee_{i \in I_1 \cup I_2} \ell_i$. The point is that forgetting allows us to equate the information obtained along two different branches and thus to merge them into a single new vertex. A sink of the branching d -program must be labelled with the negation of a d -DNFs from Σ . Thus the branching d -program is supposed by default to solve the *Search problem for Σ* : given an assignment of the variables, find a d -DNF which is falsified under this assignment.

The equivalence between a $\text{Res}(d)$ refutation of Σ and a branching d -program of the kind above is obvious. If we do not allow the forgetting of information, we will not be able to merge distinct branches, so what we get is a class of decision trees that correspond precisely to the tree-like version of these refutation systems. Indeed, a tree-like branching 2-program is depicted later in the paper in Figure 1. Naturally, if we allow querying single variables only, we get branching 1-programs—decision DAGs—that correspond to Resolution. These decision DAGs permit the view of Resolution as a game between a Prover and Adversary (originally due to Pudlak in [22]). Playing from the unique source, Prover questions variables and Adversary answers either that the variable is true or false (different plays of Adversary produce the DAG). Internal nodes are labelled by

conjunctions of facts (*records* to Pudlak) and the sinks hold conjunctions that contradict an initial clause. Prover may also choose to forget information at any point—this is the reason we have a DAG and not a tree. Of course, Prover is destined to win any play of the game—but a good Adversary strategy can force that the size of the decision DAG is large, and many Resolution lower bounds have been expounded this way.

In order to prove our lower bounds on Resolution refutations, we will use the well-known and classical technique of *bottleneck counting*. This was introduced by Haken in his seminal paper [17] (for the modern treatment see [22]). We first define the concept of *big clause*. We then design random restrictions, so that they “kill” (i.e. evaluate to \top) any big clause with high probability (w.h.p.). By the union bound, if there are few big clause, there is a restriction which kills them all. We now consider the restricted set of clauses, and using the Prover-Adversary game, show that there has to be at least one big clause in the restricted proof, which is a contradiction that completes the argument.

The case of $\text{Res}(d)$ is not so easy. A general method for proving lower bounds is developed in [25]. We first hit the refutation by random restrictions, such that all the d -DNFs in the refutation, under the restrictions, can be represented by shallow Boolean decision trees w.h.p. We then use the fact, proved in [25], that such a refutation can be transformed into a small width Resolution refutation. Finally we consider the restricted set of clauses, and using the Prover-Adversary game, show that there has to be at least one big clause in the Resolution refutation. This gives the desired contradiction to the assumption that the initial $\text{Res}(d)$ refutation contains a small number of d -DNFs.

The case of tree-like refutations, either Resolution or $\text{Res}(d)$, is much simpler, as a tree-like refutation of a given set of clause is equivalent to a decision tree, solving the search problem. We can then use a quite straightforward adversary argument against a decision tree, in order to show that it has to have many nodes. Adversary will play to a strategy that occasionally permits him to give Prover a free choice, this allows the branching in a subtree that gives a lower bound on that for the refutation.

2.3. Parameterized refutation systems

A *parameterized language* is a language $L \subseteq A^* \times \mathbb{N}$ where A is an alphabet; in an instance $(x, k) \in L$, we refer to k as the *parameter*. A parameterized language is *fixed-parameter tractable* (fpt and in FPT) if membership in L can be decided in time $f(k)|x|^{O(1)}$ for some computable function f . If FPT is the parameterized analog of P, then (at least) an infinite chain of classes vie for the honour to be the analog of NP. The so-called W-hierarchy sits thus: $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{W}[\text{SAT}]$. For more on parameterized complexity and its theory of completeness, we refer the reader to the monographs [14, 16]. Recall that the *weight* of an assignment to a propositional formula is the number of variables evaluated to true. Of particular importance to us is the parameterized problem BOUNDED-CNF-SAT whose input is (\mathcal{F}, k) where \mathcal{F} is a formula in CNF and whose yes-instances are those for which there is a satisfying assignment of weight $\leq k$. BOUNDED-CNF-SAT is complete for the class $\text{W}[2]$, and its complement

(modulo instances that are well-formed formulae) PCON is complete for the class co-W[2]. Thus, PCON is the language of *parameterized contradictions*, (\mathcal{F}, k) s.t. \mathcal{F} is a CNF which has no satisfying assignment of weight $\leq k$. The letter k is reserved for use as pertaining to this weight bound.

A *proof system* for a parameterized language $L \subseteq A^* \times \mathbb{N}$ is a poly-time computable function $P : A^* \rightarrow A^* \times \mathbb{N}$ s.t. $\text{range}(P) = L$. P is *fpt-bounded* if there exists a computable function f so that each $(x, k) \in L$ has a proof of size at most $f(k)|x|^{O(1)}$. These definitions come from [3, 4, 6] and are slightly different from those in [12, 13] (they are less unwieldy and have essentially the same properties). The program of *parameterized proof complexity* is an analog of that of Cook-Reckow [8], in which one seeks to prove results of the form $\text{W}[2] \neq \text{co-W}[2]$ by proving that parameterized proof systems are not fpt-bounded. This comes from the observation that there is an fpt-bounded parameterized proof system for a co-W[2]-complete L if $\text{W}[2] = \text{co-W}[2]$.

The system of *parameterized Resolution* [12] seeks to refute the parameterized contradictions of PCON. Given (\mathcal{F}, k) , where \mathcal{F} is a CNF in variables x_1, \dots, x_n , it does this by providing a Resolution refutation of

$$\mathcal{F} \cup \{\neg x_{i_1} \vee \dots \vee \neg x_{i_{k+1}} : 1 \leq i_1 < \dots < i_{k+1} \leq n\}. \quad (1)$$

Thus, in parameterized Resolution we have built-in access to these additional clauses of the form $\neg x_{i_1} \vee \dots \vee \neg x_{i_{k+1}}$, but we only count those that appear in the refutation. We may consider any refutation system as a parameterized refutation system, by the addition of the clauses given in (1). In particular, parameterized $\text{Res}(d)$, p- $\text{Res}(d)$, will play a part in the sequel.

3. Relativized Least number principle and $\text{Res}(d)$

The *Least number principle*, states that a (partial) order, defined on a finite set of n elements, has a minimal element. Its negation LNP can be expressed as the following FO sentence:

$$\begin{aligned} & ((\forall x \neg L(x, x)) \wedge \\ & (\forall x, y, z (L(x, y) \wedge L(y, z)) \rightarrow L(x, z)) \wedge \\ & (\forall x \exists y L(y, x))) \end{aligned} \quad (2)$$

Here $L(x, y)$ stands for $x < y$. The encoding of LNP_n as a set of clauses is as follows.

$$\begin{aligned} & \neg L_{i,i} & i \in [n] \\ & \neg L_{i,j} \vee \neg L_{j,\ell} \vee L_{i,\ell} & i, j, \ell \in [n] \\ & \bigvee_{i \in [n]} S_{i,j} & j \in [n] \\ & \neg S_{i,j} \vee L_{i,j} & i, j \in [n] \end{aligned}$$

where S is the Skolem relation, witnessing the existential variable y from $\forall x \exists y L(y, x)$, i.e., for each j , $S_{i,j} = \top$ implies that the i th element is smaller than the j th one. Of course, this S relation is unnecessary in the standard LNP_n (one may remove it and replace each $\bigvee_{i \in [n]} S_{i,j}$ with $\bigvee_{i \in [n]} L_{i,j}$). However, it will become necessary in the relativizations which we now introduce.

The negation of the d -Relativized Least number principle, $d\text{-RLNP}_n$, is as follows. Let R^p , $1 \leq p \leq d$, be the unary predicates which we relativize by, and let us denote by $\mathcal{R}(x)$ the conjunction $\bigwedge_{p \in [d]} R^p(x)$. $d\text{-RLNP}_n$ is the following sentence:

$$\begin{aligned} & ((\forall x \mathcal{R}(x) \rightarrow \neg L(x, x)) \wedge \\ & (\forall x, y, z \mathcal{R}(x) \wedge \mathcal{R}(y) \wedge \mathcal{R}(z) \rightarrow \\ & (L(x, y) \wedge L(y, z)) \rightarrow L(x, z)) \wedge \\ & (\forall x \exists y \mathcal{R}(x) \rightarrow (\mathcal{R}(y) \wedge L(y, x))) \wedge \\ & (\exists x \mathcal{R}(x))) . \end{aligned}$$

What this is saying is that the negation of the least number principle holds on the subuniverse given by $\mathcal{R}(x)$, and this subuniverse is non-empty. The corresponding translation into clauses (simplified by assuming the witness to the final \mathcal{R} be n) gives the following.

$$\begin{array}{ll} \neg \mathcal{R}_i \vee \neg L_{i,i} & i \in [n] \\ \neg \mathcal{R}_i \vee \neg \mathcal{R}_j \vee \neg \mathcal{R}_\ell \vee \neg L_{i,j} \vee \neg L_{j,\ell} \vee L_{i,\ell} & i, j, \ell \in [n] \\ \bigvee_{i \in [n]} S_{i,j} & j \in [n] \\ \neg S_{i,j} \vee \neg \mathcal{R}_j \vee R_i^p & i, j \in [n]; p \in [d] \\ \neg S_{i,j} \vee \neg \mathcal{R}_j \vee L_{i,j} & i, j \in [n] \\ R_n^p & p \in [d] \end{array}$$

We generally write RLNP_n for 1-RLNP_n .

3.1. The upper bound: $d\text{-RLNP}_n$ is easy for $\text{Res}(d+1)$

Proposition 1. *There is an $O(dn^3)$ size $\text{Res}(d+1)$ refutation of $d\text{-RLNP}_n$.*

PROOF. As in the previous subsection, \mathcal{R}_i is the d -conjunction $\bigwedge_{p \in [d]} R_i^p$; clearly $\neg \mathcal{R}_i$ is then a d -disjunction.

The $\text{Res}(d+1)$ proof will consist of n stages. We will show how to construct it, starting from the n th stage, and going to the 1st one.

The ℓ th stage clauses that we will need to derive are

$$\neg \mathcal{R}_j \vee \bigvee_{i \in [\ell], j \neq i} (L_{i,j} \wedge \mathcal{R}_i) \quad j \in [\ell] \quad (3)$$

together with

$$\bigvee_{i \in [\ell]} \mathcal{R}_i. \quad (4)$$

Thus the 1st stage clauses are $\neg \mathcal{R}_1$ and \mathcal{R}_1 which, resolved, give the empty clause. An n th stage clause of the form (3) can be derived from the axioms. We first use $d(n-1)$ applications of \wedge -introductions to derive $\neg S_{i,j} \vee \neg \mathcal{R}_j \vee \vee (\mathcal{R}_i \wedge L_{i,j})$, $j \in [n]$, $j \neq i$, and then $n-1$ resolutions of variables $S_{i,j}$. Finally we “kill” the literal $S_{i,i}$ by two resolutions with the axioms $\neg S_{j,j} \vee \neg \mathcal{R}_j \vee L_{j,j}$ and $\neg \mathcal{R}_j \vee \neg L_{j,j}$.

We derive the n th stage clause of the form (4) by directly weakening the axioms \mathcal{R}_n .

What remains is to show how to derive the $(\ell-1)$ th stage clauses from the ℓ th stage ones. The clause

$$\neg \mathcal{R}_j \vee \bigvee_{i \in [\ell-1], j \neq i} (L_{i,j} \wedge \mathcal{R}_i), \quad j \in [\ell-1] \quad (5)$$

can be derived as follows. We start off with the ℓ th stage clause

$$\neg \mathcal{R}_\ell \vee \bigvee_{i \in [\ell-1]} (L_{i,\ell} \wedge \mathcal{R}_i),$$

where we can assume the big disjunction omits $i = \ell$ by Resolution with the irreflexivity axiom $\neg \mathcal{R}_\ell \vee \neg L_{\ell,\ell}$, and manipulate it at $\ell-1$ substages, the i th substage dealing with the conjunction $L_{i,\ell} \wedge \mathcal{R}_i$, $i \in [\ell-1]$. Let us consider the clause before the i th substage, and denote it by

$$\mathcal{C} \vee (L_{i,\ell} \wedge \mathcal{R}_i) \quad (6)$$

(here \mathcal{C} is the corresponding subclause). We first resolve it with the transitivity axiom

$$\neg \mathcal{R}_i \vee \neg \mathcal{R}_\ell \vee \neg \mathcal{R}_j \vee \neg L_{i,\ell} \vee \neg L_{\ell,j} \vee L_{i,j}$$

to get

$$\neg \mathcal{R}_j \vee \neg \mathcal{R}_\ell \vee \neg L_{\ell,j} \vee \mathcal{C} \vee L_{i,j},$$

and then apply a \wedge -introduction with $\mathcal{C} \vee \mathcal{R}_i$, which is a weakening of (6) to get

$$\neg \mathcal{R}_j \vee \neg \mathcal{R}_\ell \vee \neg L_{\ell,j} \vee \mathcal{C} \vee (L_{i,j} \wedge \mathcal{R}_i).$$

Thus after having completed the $\ell-1$ substages we get the clause

$$\neg \mathcal{R}_j \vee \neg \mathcal{R}_\ell \vee \neg L_{\ell,j} \vee \bigvee_{i \in [\ell-1]} (L_{i,j} \wedge \mathcal{R}_i).$$

A resolution step with the irreflexivity axiom $\neg \mathcal{R}_j \vee \neg L_{j,j}$ “kills” the conjunction $L_{j,j} \wedge \mathcal{R}_j$, and we get

$$\neg \mathcal{R}_j \vee \neg \mathcal{R}_\ell \vee \neg L_{\ell,j} \vee \bigvee_{i \in [\ell-1], i \neq j} (L_{i,j} \wedge \mathcal{R}_i).$$

A final cut with another ℓ th stage clause,

$$\neg \mathcal{R}_j \vee \bigvee_{i \in [\ell], j \neq i} (L_{i,j} \wedge \mathcal{R}_i),$$

now gives the desired result (5).

The $(\ell - 1)$ th stage clause $\bigvee_{i \in [\ell-1]} \mathcal{R}_i$ is easier to derive. We weaken the ℓ th stage clause $\neg \mathcal{R}_\ell \vee \bigvee_{i \in [\ell-1]} (L_{i,\ell} \wedge \mathcal{R}_i)$ $\ell - 1$ times to get $\neg \mathcal{R}_\ell \vee \bigvee_{i \in [\ell-1]} \mathcal{R}_i$. We then resolve it with the ℓ th stage clause $\bigvee_{i \in [\ell]} \mathcal{R}_i$.

This completes the proof.

3.2. An optimal lower bound: $RLNP_n$ is exponentially hard for Resolution

We will prove the following using the well-known method of random restrictions.

Proposition 2. *Any Resolution proof of $RLNP_n$ is of size $2^{\Omega(n)}$.*

PROOF. The idea is to randomly divide the universe \mathbb{U} into two approximately equal parts. One of them, \mathbb{R} , will represent the predicate R ; all the variables within it will remain unset. The rest, \mathbb{C} , will be the “chaotic” part; all the variables within \mathbb{C} and most of the variables between \mathbb{C} and \mathbb{R} will be set at random. It is now intuitively clear that while \mathbb{C} kills with positive probability a certain number of “big” clauses, \mathbb{R} allows to show, via an adversary argument, that at least one such clause must be present in any Resolution refutation, after it has been hit by the random restrictions. Therefore a huge number of “big” clauses must have been presented in the original refutation.

We of course keep $R_n := \top$. The *random restrictions* are as follows.

1. We first set all the variables $R_i, i \in [n - 1]$, to either \top or \perp independently at random with equal probabilities, $1/2$. Let us denote the set of variables with $R_i = \top$ by \mathbb{R} , and the set of variables with $R_i = \perp$ by \mathbb{C} , $\mathbb{C} = \mathbb{U} \setminus \mathbb{R}$.
2. We now set all the variables $L_{i,j}$ with at least one endpoint in \mathbb{C} , i.e. $\{i, j\} \cap \mathbb{C} \neq \emptyset$, to either \top or \perp independently at random with equal probabilities, $1/2$.
3. For each $j \in \mathbb{C}, j \neq i$ we set $S_{i,j}$ to either \top or \perp independently at random with equal probabilities $1/2$. Note that it is possible to set all the $S_{i,j}$ to \perp , thus violating an axiom. It however happens with small probability, $1/2^{n-1}$ for a fixed j .
4. We finally set all the variables $S_{i,j}$ with $i \in \mathbb{C}, j \in \mathbb{R}$ to \perp .

Note the unset variables define exactly the non-relativized principle on \mathbb{R} , $LNP_{|\mathbb{R}|}$.

By the Chernoff bound (see [21]) the probability that \mathbb{R} contains less than $n/4$ elements is exponentially small, and we therefore have the following.

Observation 1. *The probability that the random restrictions are inconsistent (i.e. violate an axiom) or $|\mathbb{R}| \leq n/4$ is at most $(n-1)2^{-(n-1)} + e^{-n/16}$.*

A *big clause* is one which contains at least $n/8$ busy elements. The element i is *busy* in the clause \mathcal{C} iff \mathcal{C} contains one of the following variables, either positively or negatively: $R_i, L_{i,j}, L_{j,i}, S_{j,i}$ for some $j, j \neq i$ (note the omission of $S_{i,j}$). We can now deduce the following.

Observation 2. *A clause, containing p busy elements, does not evaluate to \top under the random restrictions with probability at most $(3/4)^{p/2}$.*

Indeed, let us consider the different cases of a busy element i in the clause \mathcal{C} :

1. The variable R_i is present in \mathcal{C} : The probability that the corresponding term does not evaluate to \top is $1/2$.
2. The variable $S_{j,i}$ for some $i \neq j$ is present in \mathcal{C} : The corresponding term does not evaluate to \top if either $i \in \mathbb{R}$, or $i \in \mathbb{C}$ and it evaluates to \perp . The probability of this is $3/4$.
3. Either the variable $L_{i,j}$ or the variable $L_{j,i}$ for some $j \neq i$ is present in \mathcal{C} . Let us denote the set of all such elements i or j by \mathbb{V} , $|\mathbb{V}| = \ell$, and the corresponding subclause of \mathcal{C} induced by those elements as \mathcal{E} . That is, \mathcal{E} contains precisely those atoms of \mathcal{C} that are of the form $L_{i,j}$ or $L_{j,i}$. Construct the graph \mathbb{G} with vertex set \mathbb{V} and edge set \mathbb{E} determined by the variables $L_{i,j}$, i.e. $\mathbb{E} = \{\{i,j\} \mid L_{i,j} \text{ is present in } \mathcal{C}\}$. Consider any spanning forest of \mathbb{G} . Assume that all the roots are in \mathbb{R} as this only increases the probability that \mathcal{E} does not evaluate to \top . Going from the root to the leaves in each tree, we see that the probability that the corresponding edge does not evaluate to \top is $3/4$ (the same reason as in the 2nd case). Moreover all the edge variables are independent from each other, and also there are at most $\ell/2$ roots (exactly $\ell/2$ iff the forest consists of trees having a root and a single leaf only). Therefore the probability that the subclause \mathcal{E} does not evaluate to \top is at most $(3/4)^{\ell/2}$.

As the events from 1, 2 and 3 are independent for different elements from \mathbb{U} , we have completed the argument for Observation 2.

We can now present the main argument in the proof. We recall that a big clause is one which contains at least $n/8$ busy elements. Assume there is a Resolution refutation of RLNP_n which contains less than $(4/3)^{n/8}$ big clauses. From the Observations 1 and 2, using the union-bound on probabilities, we can conclude that there is a restriction which is consistent, “kills” all the big clauses (evaluating them to \top), and leaves \mathbb{R} big enough ($|\mathbb{R}| \geq n/4$). This is because $(n-1)2^{-(n-1)} + e^{-n/16} + (4/3)^{n/8}(3/4)^{n/16} < 1$. Recall that the restricted refutation is nothing but a Resolution refutation of $\text{LNP}_{|\mathbb{R}|}$ on \mathbb{R} . What remains to show is that any such refutation must contain a big clause which would contradict to the assumption there were “few” big clauses in the original refutation.

We will consider the Prover-Adversary game for $\text{LNP}_{|\mathbb{R}|}$. At any time \mathbb{R} is represented as a disjoint union of three sets, $\mathbb{R} = \mathbb{B} \uplus \mathbb{W} \uplus \mathbb{F}$. \mathbb{B} is the set

of all the elements busy in the current clause. The elements of \mathbb{B} are always totally ordered. \mathbb{W} is the set of witnesses for some elements in \mathbb{B} , i.e. for each $j \in \mathbb{B}$ there is an element $i \in \mathbb{W} \uplus \mathbb{B}$ such that $S_{i,j} = \top$. We assume that, at any time, any element of \mathbb{W} is smaller than all the elements of \mathbb{B} . \mathbb{F} is the set of “free” elements. It is obvious how Adversary maintains these sets in the Prover-Adversary game. When a variable, which makes an element $i \in \mathbb{W} \uplus \mathbb{F}$ busy, is queried he adds i at the bottom of the totally ordered set \mathbb{B} , answers accordingly, and chooses some $j \in \mathbb{F}$ and moves it to \mathbb{W} setting $S_{j,i} = \top$. When all the variables, which kept an element \mathbb{B} busy, are forgotten, Adversary removes i from \mathbb{B} and removes the corresponding witness j from \mathbb{W} if it is there (note that it may be in \mathbb{B} , too, in which case it is not removed). In this way Adversary can maintain the partial assignment consistent as far as $\mathbb{F} \neq \emptyset$. Note also that $|\mathbb{B}| \geq |\mathbb{W}|$. Therefore at the moment a contradiction is reached we have $|\mathbb{B}| \geq |\mathbb{R}|/2 \geq n/8$ as claimed.

3.3. General lower bounds: $d\text{-RLNP}_n$ is subexponentially hard for $\text{Res}(d)$

We will first give the necessary background from [25].

Definition 1 (Definition 3.1, [25]). A decision tree is a rooted binary tree in which every internal node queries a propositional variable, and the leaves are labelled by either \top or \perp .

Thus every path from the root to a leaf may be viewed as a partial assignment. Let us denote by $Br_v(\mathfrak{T})$, for $v \in \{\top, \perp\}$, the set of paths (partial assignments) in the decision tree \mathfrak{T} which lead from the root to a leaf labelled by v .

A decision tree \mathfrak{T} strongly represents a DNF \mathcal{F} iff for every $\pi \in Br_v(\mathfrak{T})$, $\mathcal{F} \upharpoonright_{\pi} = v$.

The representation height of \mathcal{F} , $h(\mathcal{F})$, is the minimum height of a decision tree strongly representing \mathcal{F} .

Definition 2 (Definition 3.2, [25]). Let \mathcal{F} be a DNF, and \mathbb{S} be a set of variables. We say that \mathbb{S} is a cover of \mathcal{F} iff every conjunction of \mathcal{F} contains a variable from \mathbb{S} . The **covering number** of \mathcal{F} , $c(\mathcal{F})$, is the minimum size of a cover of \mathcal{F} .

Lemma 1 (Corollary 3.4, [25]). Let $d \geq 1$, $\alpha > 0$, $1 \geq \beta, \gamma > 0$, $s > 0$, and let \mathcal{D} be a distribution on partial assignments such that for every $d\text{-DNF}$ \mathcal{G} , $Pr_{\rho \in \mathcal{D}}[\mathcal{G} \upharpoonright_{\rho} \neq \top] \leq \alpha 2^{-\beta(c(\mathcal{G}))^\gamma}$. Then for every $d\text{-DNF}$ \mathcal{F} :

$$Pr_{\rho \in \mathcal{D}}[h(\mathcal{F} \upharpoonright_{\rho}) \geq s] \leq \alpha d 2^{-2(\beta/4)^d (s/2)^{\gamma^d}}.$$

Lemma 2 (Theorem 5.1, [25]). Let \mathcal{G} be a set of clauses of width at most w . If \mathcal{G} has a $\text{Res}(d)$ refutation so that for each line \mathcal{L} of the refutation, of Γ , $h(\mathcal{L}) \leq w$, then \mathcal{G} has a Resolution refutation of width at most dw .

We also need the following construction.

Lemma 3 (Subsections 8.3 and 8.4 in [25]). *There is an undirected graph $\mathfrak{G}([n], E)$ on n vertices and max-degree $\theta(\ln n)$ such that any Resolution refutation of LNP_n , restricted on \mathfrak{G} , is of width $\Omega(n)$.*

LNP_n , restricted on \mathfrak{G} , means that for each element i the witness j has to be a neighbour of i in \mathfrak{G} , i.e. we set $S_{i,j} = \perp$ whenever $\{i, j\} \notin E$.

We can now prove the desired result.

Proposition 3. *For every constant $d \geq 1$ there is a constant $\varepsilon_d \in (0, 1]$ such that any $\text{Res}(d)$ refutation of $d\text{-RLNP}_n$ is of size $2^{\Omega(n^{\varepsilon_d})}$.*

PROOF. We again denote by \mathcal{R}_i the d -conjunction $\bigwedge_{p \in [d]} R_i^p$. We consider $d\text{-RNLP}_n$, restricted on the graph \mathbb{G} from Lemma 3. Thus we have eliminated all the “big” axioms in the encoding of $d\text{-RNLP}_n$; the biggest ones are now of width $\theta(\ln n)$, and therefore it would be possible to eventually apply Lemma 2.

The *random restrictions* are very similar to the ones in the proof of Proposition 2. We fix \mathcal{R}_n (i.e. $R_n^p = \top$ for all $p \in [d]$) to \top .

1. For each $i \in [n-1]$ and $p \in [d]$ we set the variable R_i^p to either \top or \perp , independently at random with equal probabilities, $1/2$. We denote the set of elements with $\mathcal{R}_i = \top$ by \mathbb{R} , and the rest by \mathbb{C} , $\mathbb{C} = \mathbb{U} \setminus \mathbb{R}$. Note that $\mathbb{R} \neq \emptyset$ as always $n \in \mathbb{R}$, and by the Chernoff bound

$$\text{Prob} \left[|\mathbb{R}| \leq \frac{n}{2^{d+1}} \right] \leq e^{-n/2^{d+3}}.$$

2. We now set all the variables $L_{i,j}$ with at least one endpoint in \mathbb{C} , i.e. $\{i, j\} \cap \mathbb{C} \neq \emptyset$, to either \top or \perp independently at random with equal probabilities, $1/2$.
3. For each $j \in \mathbb{C}$, and i , a neighbour of j in \mathbb{G} , we set $S_{i,j}$ to either \top or \perp independently at random with equal probabilities $1/2$. Note that it is possible to set all the $S_{i,j}$ to \perp , thus violating an axiom. It happens with probability $1/2^{\deg_{\mathfrak{G}}(i)} = 1/n^{\omega_0}$ where ω_0 is the constant, hidden in θ -denotation in Lemma 3, and note that we can choose $\omega_0 \geq 1$.
4. We finally set all the variables $S_{i,j}$ with $j \in \mathbb{R}$, $i \in \mathbb{C}$ to \perp .

The unset variables define exactly the non-relativized principle on \mathbb{R} over \mathfrak{G} , $LNP_{|\mathbb{R}|}$, and we have

Observation 3. *The probability that the random restrictions are inconsistent (i.e. violate an axiom) or $|\mathbb{R}| \leq n/2^{d+1}$ is at most $1/n^{\omega_0-1} + e^{-n/2^{d+3}}$.*

We will now consider the effect of the random restrictions on a given d -DNF \mathcal{C} . We need a few definitions first: We say that a variable R_i^p for some $p \in [d]$ *mentions* the element i ; a variable $L_{i,j}$ mentions both i and j ; a variable $S_{i,j}$ mentions j only (cf. definition of business in Proposition 2). A formula mentions the union of elements mentioned by some variable from the formula. The *element-cover number* of a d -DNF \mathcal{F} , $c'(\mathcal{F})$, is the minimum cardinality of a set of elements, such that each element is mentioned by at least one term of the clause. There is an obvious connection between $c'(\mathcal{F})$ and the cover $c(\mathcal{F})$:

Observation 4. $c'(\mathcal{F}) = \Omega\left(\sqrt{c(\mathcal{F})}/\log n\right)$.

Indeed m elements mention dm variables R_i^p , and at most $\binom{m}{2} + m \max_i \deg_{\mathbb{G}}(i)$ variables $L_{i,j}/S_{i,j}$ which makes $O(m^2 + m \log n)$ in total, and therefore $c(\mathcal{F}) = O\left((c'(\mathcal{F}))^2 + c'(\mathcal{F}) \log n\right)$.

We can now show that any d -DNF \mathcal{F} collapses under random restrictions to a short decision tree w.h.p. Let us first note that \mathcal{F} mentions at most $\omega_0 \log n$ variables $S_{j,n}$. We build up a decision tree by first querying all these. This contributes $\omega_0 \log n$ to the height and there are at most n^{ω_0} leaves which contains d -DNFs which do not mention the n th element.

Let us take such a d -DNF \mathcal{G} . We are now going to use Lemma 1. Let us perform the following experiment. Take any term, \mathcal{T} , of \mathcal{G} . Each literal of \mathcal{T} , containing a variable R_i^p , evaluates to \top with probability $1/2$. Each literal, containing a variable $S_{j,i}$, has a probability for $i \in \mathbb{C}$ at least $1/2$; indeed the rest of \mathcal{T} may contain at most $d-1$ positive appearances of variables R_i^p and then the last one, not in \mathcal{T} , “decides” $i \in \mathbb{C}$ with probability $1/2$. Therefore the literal, containing a variable $S_{j,i}$, evaluates to \top with probability at least $1/4$. The same argument applies to a literal, containing a variable $L_{i,j}$. Thus the probability that \mathcal{T} evaluates to \top under the random restrictions is at least $1/4^d$ (the fact that the term contains at most d variables is essential here; indeed consider the $d+1$ -term $\mathcal{R}_i \wedge S_{j,i}$: $\mathcal{R}_i = \top$ enforces $S_{j,i}$ either \perp or unset, and therefore there is no way to evaluate the term to \top). On the other hand \mathcal{T} mentions at most $2d$ elements, so we can repeat the above procedure (i.e. picking a new term) at least $c'(\mathcal{G})/(2d)$ times and the probability that each term does not evaluate to \top is at most $1 - 1/4^d$. Moreover these $c'(\mathcal{G})/(2d)$ trials are independent as in each of them only elements, not mentioned so far, are involved. Therefore the probability that \mathcal{G} does not evaluate to \top under the random restrictions is at most $(1 - 1/4^d)^{c'(\mathcal{G})/(2d)}$ which is at most $(1 - 1/4^d)^{\sqrt{c(\mathcal{G})}/(2d \log n)}$ by Observation 4.

We can now apply Lemma 1 with $\alpha = 1$, $\beta = \omega_1/\log n$, where ω_1 is a constant, depending on d only, and $\gamma = 1/2$. We set $s = \omega_2 n$, where ω_2 will be fixed later, and will depend on d only. What we get by the lemma is that the probability that \mathcal{G} , under the random restrictions, cannot be represented by a decision tree of height at most $\omega_2 n$ is at most $\exp\left(-\omega_3 n^{1/2^d}/(\log n)^d\right)$, where ω_3 is a constant, dependent on d . Going back to the initial d -DNF \mathcal{F} , we see that the probability \mathcal{F} , under the random restrictions, cannot be represented by a decision tree of height at most $\omega_2 n + \omega_0 \log n$ is at most $n^{\omega_0} \exp\left(-\omega_3 n^{1/2^d}/(\log n)^d\right)$.

We can finally present the main argument of the proof. Suppose, for the sake of contradiction, that there is a $\text{Res}(d)$ refutation of $d\text{-RLNP}_n$ containing less than $n^{-\omega_0} \exp\left((\omega_3/2) n^{1/2^d}/(\log n)^d\right)$ d -DNFs. By the union bound, the probability that at least one of them, under the random restrictions, cannot be represented by a decision tree of height at most $\omega_2 n + \omega_0 \log n$ is at

most $\exp\left(-(\omega_3/2) n^{1/2^d} / (\log n)^d\right)$. By Lemma 2 then, with the same probability, the restricted $\text{Res}(d)$ refutation cannot be transformed to a width $\omega_2 dn + \omega_0 d \log n$ Resolution refutation. Adding up this probability with the probability from Observation 3, we see there exists a “good” restriction, i.e. such that the final Resolution refutation is of width $\omega_2 dn + \omega_0 d \log n$. Recall that we are still free to choose ω_2 (it affects ω_3 , but it is fine as ω_3 is a constant, dependent on d , too) and it depends on d only, so we can ensure that ω_2 is smaller than the constant, hidden in the Ω -notation in Lemma 3, divided by 2^{d+1} (recall that the size of \mathbb{R} is at least $n/2^{d+1}$). This is the desired contradiction which completes the proof.

4. Relativized Induction principle and $\text{Res}^*(d)$

In this section we consider a version of the Induction principle, denoted IP_n , that can be encoded as an FO sentence if a built-in predicate, defining a total order on the universe, is added to the language. It is easy to show that IP_n is easy for tree-like Resolution, and so it is also for its d th relativization $d\text{-RIP}_n$, but for $\text{Res}^*(d+1)$. Finally we prove that $d\text{-RIP}_n$ is hard for $\text{Res}^*(d)$.

4.0.1. Induction principle

The (negation of the) Induction principle, we consider, is the following simple statement: Given an ordered universe, there is a property P , such that

1. The property holds for the smallest element.
2. If $P(x)$ hold for some x , then there is y , bigger than x , and such that $P(y)$ holds.
3. The property does not hold for the biggest element.

The universe \mathbb{U} can now be considered as the set of the first n natural numbers. In our language we can use the relation symbol $<$ with its usual meaning. We can also use any constant c as well as $n - c$ (note that in the language n denotes the maximal element of \mathbb{U} , while 1 denotes the minimal one). The Induction principle, we have just described, can be written as

$$P(1) \wedge \forall x \exists y ((x < y \wedge P(x)) \rightarrow P(y)) \wedge P(n).$$

The translation into propositional logic gives the following set of clauses

$$\begin{aligned} & P_1, \neg P_n \\ & \bigvee_{j=i+1}^n S_{i,j} \quad i \in [n-1] \\ & \neg S_{i,j} \vee \neg P_i \vee P_j \quad i, j \in [n], i < j. \end{aligned}$$

The relativized version's translation is

$$\begin{aligned} &P_1, \neg P_n \\ &R_1^p, R_n^p \quad p \in [d] \\ &\bigvee_{j=i+1}^n S_{i,j} \quad i \in [n-1] \end{aligned} \tag{7}$$

$$\neg S_{i,j} \vee \neg \mathcal{R}_i \vee R_j^p \quad i, j \in [n], i < j, p \in [d] \tag{8}$$

$$\neg S_{i,j} \vee \neg \mathcal{R}_i \vee \neg P_i \vee P_j \quad i, j \in [n], i < j. \tag{9}$$

4.1. The upper bound

Proposition 4. *There is an $O(dn^2)$ size $\text{Res}^*(d+1)$ refutation of $d\text{-RIP}_n$.*

PROOF. We first apply \wedge -introduction between the clauses (8) and (9), d times, to get the clauses

$$\neg S_{i,j} \vee \neg \mathcal{R}_i \vee \neg P_i \vee (\mathcal{R}_j \wedge P_j) \quad 1 \leq i < j \leq n.$$

For every i we resolve these with the clauses (7) to get

$$\neg \mathcal{R}_i \vee \neg P_i \vee \bigvee_{j \geq i+1} (\mathcal{R}_j \wedge P_j). \tag{10}$$

The i th stage clause is now

$$\bigvee_{j \geq i+1} (\mathcal{R}_j \wedge P_j). \tag{11}$$

For $i = 1$ it is derived by resolving (10) with the pure-literal axioms P_1 and R_1^p , $p \in [d]$. The induction step is pretty easy: we resolve the $(i-1)$ th stage clause $\bigvee_{j \geq i} (\mathcal{R}_j \wedge P_j)$ with the clause (10) to get the i th stage clause (11). After the $(n-1)$ th stage we have derived the pure-term clause $\mathcal{R}_n \wedge P_n$. We now weaken it to P_n and then resolve it with the axiom $\neg P_n$ to get the desired empty clause.

The number of resolution steps is $O(dn^2)$. At each stage we have resolved the clause obtained at the previous stage only once, therefore the $\text{Res}(d+1)$ refutation, we have constructed, is tree-like.

4.2. The lower bound

We will first prove it for $d = 1$, i.e. that RIP_n is exponentially hard for tree-like Resolution. We will then generalize it to any d .

Proposition 5. *Any tree-like Resolution refutation of RIP_n is of size $2^{\Omega(n)}$.*

PROOF. We will use an adversary strategy against the decision tree solving the search problem.

We say that the variables P_i , R_i and $S_{i,j}$ for $j > i$ are *associated* to the i th element. When one of these has been queried for the first time by Prover,

Adversary fixes all of them, so that the i th element becomes *busy* (cf. definition of business in Section 3). Initially only the minimal and maximal elements are busy as the singleton clauses P_1 , R_1 , $\neg P_n$ and R_n force the values of the corresponding variables. For technical reasons only, we assume that the $(n-1)$ th element is busy too, by setting $S_{n-1,n} = \top$, $R_{n-1} = \top$ and $P_{n-1} = \perp$. The elements that are not busy we call *free*, with the exception of those below the *source*. The source is the biggest element j , such that $R_j = P_j = \top$. Initially the source is the first element. It is important to note that no contradiction can be found as far as there is at least one free element bigger than the source. All the variables associated to the elements smaller than the source are set (consistently with the axioms) in the current partial assignment. Thus there are free elements only between the source and the maximal element. Informally speaking, Prover's strategy is moving the source towards the end of the universe, the $(n-2)$ th element.

We will prove that at any stage in the Prover-Adversary game, the number of free elements can be used to lower bound the subtree, rooted at the current node of the tree. More precisely, if $T(m)$ is the size of the subtree rooted at a node, where there are m such elements, we will show that $T(m) \geq \varphi_m$. Here φ_m is the m th Fibonacci number, defined by

$$\begin{aligned}\varphi_0 &= \varphi_1 = 1 \\ \varphi_m &= \varphi_{m-1} + \varphi_{m-2} \quad \text{for } m \geq 2.\end{aligned}$$

Initially, we have $n-3$ free elements bigger than the source, therefore the inequality we claim, together with the known asymptotic $\varphi_m \sim \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^m$, implies the desired lower bound.

What remains is to prove $T(m) \geq \varphi_m$. We use induction on m . The basis cases $m=0$ or $m=1$ are trivial. To prove the induction step, we consider all the possibilities for a Prover's query:

1. It is about either a busy element or an element below the source. As already explained, the value of such a variable is already known in the current partial assignment. Adversary answers; the value of m does not change.
2. The query is about a free element i , and recall that it is bigger than the source. If the variable queried is either R_i or P_i , Adversary first sets $S_{i,n} = \top$, $S_{i,j} = \perp$ for all j , $i < j < n$, and then allows Prover a free choice between the two possibilities: either $R_i = \top$, $P_i = \perp$ or $R_i = \perp$, $P_i = \top$. If the variable queried is $S_{i,j}$, for some $j > i$, Adversary first sets $R_i = P_i = \perp$, $S_{i,l} = \top$ for all $l \neq j$, and then allows Prover a free choice between either $S_{i,j} = \top$ or $S_{i,j} = \perp$. The number of free elements, m , decreases by one. Therefore we have

$$T(m) \geq 2T(m-1).$$

By the induction hypothesis $T(m-1) > \varphi_{m-1}$, and then $T(m) \geq 2\varphi_{m-1} \geq \varphi_m$.

3. The query is about the source, i.e. the variable queried is $S_{i,j}$, where i is the source's index. If the j th element is busy, Adversary answers \perp . If the j th element is free, but far away from the source, that is there are at least two free elements between the source and the j th element, Adversary answers \perp , too. Neither the position of the source nor the value of m changes. The only remaining case is when the j th element is both free and near to the source, that is one of the two smallest free elements, bigger than the source. Adversary now offers Prover a free choice to move the source to any of these two elements, by giving the corresponding answer: \top —source is moved to the j th element or \perp —source is moved to the other nearest element. In one of these choices m decreases by one, and in the other it decreases by two. Therefore we have

$$T(m) \geq T(m-1) + T(m-2).$$

The induction hypothesis gives $T(m-1) \geq \varphi_{m-1}$ and $T(m-2) \geq \varphi_{m-2}$. Thus

$$T(m) \geq \varphi_{m-1} + \varphi_{m-2} = \varphi_m.$$

This completes the proof.

We will show how to modify the proof in order to prove a more general statement.

Proposition 6. *Any $\text{Res}^*(d)$ refutation of $d\text{-RIP}_n$ is of size $2^{\Omega(n/d)}$.*

PROOF. The proof is very similar to the previous one, so we will explain what changes should be made in there. Our task is now to prove that $T(m) \geq \rho_d^m$. Where ρ_d is the largest real positive root of the equation

$$x^{d+1} - x - 1 = 0.$$

It is not hard to see that

$$1 + \frac{\alpha}{d} \leq \rho_d \leq 1 + \frac{\beta}{d}$$

for some appropriately chosen constants α and β . Thus we would get the desired result as $T(m) \geq \left(1 + \frac{\alpha}{d}\right)^m \geq e^{m/(1+d/\alpha)}$ (we have used the known inequality $(1+x)^{1+1/x} > e$ for $x > 0$).

We first set the variables, associated to the last $d+1$ elements, by setting $\mathcal{R}_j = \top$ (i.e. $R_j^p = \top$ for all $p \in [d]$), $P_j = \perp$ and $S_{j,n} = \top$ for all j , $n-d \leq j \leq n-1$.

Prover now can query d -disjunctions instead of single variables. Adversary first simplifies the query, using the current partial assignment, and then answers as follows:

1. The resulting query evaluates to either \perp or \top under the current partial assignment. Adversary replies with the corresponding value. Clearly the number of free elements, m , does not change.

2. The resulting query involves only free elements; recall that they all are bigger than the source. Suppose there are d' ($d' \leq d$) such elements. Adversary allows Prover a free choice between \perp and \top , without moving the source to one of the new elements. This is because the only way to force such a movement is a positive answer to a query $\mathcal{R}_j \wedge P_j$ which is of size $d + 1$. The number of free elements decreases by d' , and therefore we have

$$T(m) \geq 2T(m - d').$$

By the induction hypothesis $T(m - d') \geq \rho_d^{m-d'} \geq \rho_d^{m-d}$, and then $T(m) \geq 2\rho_d^{m-d} > \rho_d^{m-d} + \rho_d^{m-d-1} = \rho_d^m$.

3. The resulting query involves the source, i.e. contains variable(s) of the form $S_{i,j}$, where i is the source's index and j is a free element. Denote the set of all such elements by \mathbb{J} , i.e. $\mathbb{J} = \{j \mid S_{i,j} \text{ is in the query and } j \text{ is free}\}$. If all the elements in \mathbb{J} are far away from the source, that is at distance at least $d + 2$, Adversary first sets all the $S_{i,j} \perp$ and then answers the resulting query as in Case 2. In the other case, when at least one element from \mathbb{J} is near to the source, it is always possible to move the source to two elements between 1st or $(d + 1)$ th nearest free element at worst, and the choice of which may be given to Prover. In the former case we have an instance with at least $m - d$ free elements, and in the latter with at least $m - d - 1$. This gives

$$T(m) \geq T(m - d) + T(m - d - 1).$$

The induction hypothesis gives $T(m - d) \geq \rho_d^{m-d}$ and $T(m - d - 1) \geq \rho_d^{m-d-1}$. Thus

$$T(m) \geq \rho_d^{m-d} + \rho_d^{m-d-1} = \rho_d^m.$$

This completes the proof.

5. Separating p-Res(1) and p-Res(2)

We recall RLNP_n and its salient properties of being polynomial to refute in $\text{Res}(2)$, but exponential in $\text{Res}(1)$ (as in Section 3). Polynomiality clearly transfers to fpt-boundedness in $\text{p-Res}(2)$, so we address the lower bound for $\text{p-Res}(1)$.

5.1. Lower bound: A strategy for Adversary over RLNP_n

We will give a strategy for Adversary in the game representation of a $\text{p-Res}(1)$ refutation. The argument used in Section 3 does not adapt to the parameterized case, so we instead use a technique developed specifically for the parameterized Pigeonhole principle in [6].

Recall that a *parameterized clause* is of the form $\neg v_1 \vee \dots \vee \neg v_{k+1}$ (where each v_i is some R , L or S variable). The i, j appearing in R_i , $L_{i,j}$ and $S_{i,j}$ are termed *co-ordinates*. We define the following *random restrictions*. Set $R_n := \top$.

Randomly choose $i_0 \in [n-1]$ and set $R_{i_0} := \top$ and $L_{i_0,n} = S_{i_0,n} := \top$. Randomly choose $n - \sqrt{n}$ elements from $[n-1] \setminus i_0$, and call this set \mathbb{C} . Set $R_i := \perp$ for $i \in \mathbb{C}$. Pick a random bijection π on \mathbb{C} and set $L_{i,j}$ and $S_{i,j}$, for $i, j \in \mathbb{C}$, according to whether $\pi(j) = i$. Set $L_{i,j} = L_{j,i} = S_{i,j} = S_{j,i} := \perp$, if $j \in \mathbb{C}$ and $i \in [n] \setminus (\mathbb{C} \cup \{i_0\})$.

What is the probability that a parameterized clause is **not** evaluated to true by the random assignment? We allow that each of $\neg R_n$, $\neg R_{i,0}$, $\neg L_{i_0,n}$ and $\neg S_{i_0,n}$ appear in the clause—leaving $k+1-4 = k-3$ literals, within which must appear $\sqrt{(k-3)/4}$ distinct co-ordinates. The probability that some $\neg R_i$, $i \notin \{i_0, n\}$, fails to be true is bound above by the probability that i is in $[n-1] \setminus (\mathbb{C} \cup \{i_0\})$ —which is $\leq \frac{\sqrt{n}-2}{n-2} \leq \frac{1}{\sqrt{n}}$. The probability that some $\neg L_{i,j}$ fails to be true, where one of the co-ordinates i, j is possibly mentioned before and $(i, j) \neq (i_0, n)$, is bound above by the probability that both i, j are in $[n] \setminus \mathbb{C}$ plus the probability that both i, j are in \mathbb{C} and $i = \pi(j)$. This gives the bound $\leq \frac{\sqrt{n}}{n} \cdot \frac{\sqrt{n}-1}{n-1} + \frac{n-\sqrt{n}}{n} \cdot \frac{n-\sqrt{n}-1}{n-1} \cdot \frac{1}{n-\sqrt{n}-1} \leq \frac{2}{n} \leq \frac{1}{\sqrt{n}}$. Likewise with $\neg S_{i,j}$. Thus we get that the probability that a parameterized clause is **not** evaluated to true by the random assignment is $\leq \frac{1}{\sqrt{n}} \sqrt{(k-3)/4} = n^{-\sqrt{(k-3)/16}}$.

Now we are ready to complete the proof. Suppose fewer than $n\sqrt{(k-3)/16}$ parameterized clauses appear in a p-Res(1) refutation of RLNP_n , then there is a random restriction as per the previous paragraph that evaluates all of these clauses to true. What remains is a Res(1) refutation of $\text{RLNP}_{\sqrt{n}}$, which must be of size larger than $n\sqrt{(k-3)/16}$ itself, for n sufficiently large (see [9]). Thus we have proved.

Theorem 1. *Every p-Res(1) refutation of RLNP_n is of size $\geq n\sqrt{(k-3)/16}$.*

6. Separating p-Res*(1) and p-Res*(2)

Let us recall the important properties of IP_n and RIP_n , from the perspective of Section 4. IP_n admits refutation in $\text{Res}^*(1)$ in polynomial size, as does RIP_n in $\text{Res}^*(2)$. But all refutations of RIP_n in $\text{Res}^*(1)$ are of exponential size. In the parameterized world things are not quite so well-behaved. Both IP_n and RIP_n admit refutations of size, say, $\leq 4k!$ in p-Res*(1); just evaluate variables $S_{i,j}$ from $i := n-1$ downwards. Thus ask in sequence

$$S_{n-1,n}, S_{n-2,n-1}, S_{n-2,n}, \dots, S_{n-k,n-k+1}, \dots, S_{n-k,n},$$

each level $S_{n-i,n-i+1}, \dots, S_{n-i,n}$ surely yielding a true answer. Clearly this is an fpt-bounded refutation. We are forced to consider something more elaborate, and thus we introduce the *Relativized Vectorized Induction Principle* RVIP_n below. Roughly speaking, we stretch each single level of RIP_n into n copies of

itself in RVIP_n , to make things easier for Adversary.

$$\begin{array}{ll}
R_1, P_{1,1}, R_n, \neg P_{n,j} & j \in [n] \\
\bigvee_{l>i, m \in [n]} S_{i,j,l,m} & i, j \in [n] \\
\neg S_{i,j,l,m} \vee \neg R_i \vee \neg P_{i,j} \vee R_l & i \in [n-1], j, l, m \in [n] \\
\neg S_{i,j,l,m} \vee \neg R_i \vee \neg P_{i,j} \vee P_{l,m} & i \in [n-1], j, l, m \in [n]
\end{array}$$

6.1. Lower bound: A strategy for Adversary over RVIP_n

We will give a strategy for Adversary in the game representation of a $\text{Res}^*(1)$ refutation. For convenience, we will assume that Prover never questions the same variable twice (this saves us from having to demand trivial consistencies in future evaluations).

Information conceded by Adversary of the form $R_i, \neg R_i, P_{i,j}$ and $S_{i,j,l,m}$ makes the element i *busy*. $\neg P_{i,j}$ and $\neg S_{i,j,l,m}$ do not make i busy, in the case of $\neg P_{i,j}$ this is a departure from earlier definitions of business (due to the vectorization, there are now n ways that some i can become true as $P_{i,j}$). The *source* is the largest element i for which there is a j such that Adversary has conceded $R_i \wedge P_{i,j}$. Initially, the source is 1. Adversary always answers $R_1, P_{1,1}, R_n, \neg P_{n,j}$ (for $j \in [n]$), according to the axioms. Thus $i := 1$ and n are somehow special, and the size of the set inbetween is $n - 2$. In the following, i refers to the first index of a variable.

If i is below the source. When Adversary is asked $R_i, P_{i,j}$ or $S_{i,j,l,m}$, then he answers \perp .

If i is above the source. When Adversary is asked R_i , or $P_{i,j}$, then he gives Prover a free choice unless: 1.) R_i is asked when some $P_{i,j}$ was previously answered \top (in this case R_i should be answered \perp); or 2.) Some $P_{i,j}$ is asked when R_i was previously answered \top (in this case $P_{i,j}$ should be answered \perp). When Adversary is asked $S_{i,j,l,m}$, then again he offers Prover a free choice. If Prover chooses \top then Adversary sets $P_{i,j}$ and R_i to \perp .

Suppose i is the source. Then Adversary answers $P_{i,j}$ and $S_{i,j,l,m}$ as \perp , unless $R_i \wedge P_{i,j}$ witnesses the source. If $R_i \wedge P_{i,j}$ witnesses the source, then, if l is not the next non-busy element above i , answer $S_{i,j,l,m}$ as \perp . If l is the next non-busy element above i , then give $S_{i,j,l,m}$ a free choice, unless $\neg P_{l,m}$ is already conceded by Adversary, in which case answer \perp . If Prover chooses \top for $S_{i,j,l,m}$ then Adversary sets R_l and $P_{l,m}$ to \top .

Using this strategy, Adversary can not be caught lying until either he has conceded that k variables are true, or he has given Prover at least $n - 2$ free choices.

Let $T(p, q)$ be some monotone decreasing function that bounds the size of the game tree from the point at which Prover has answered p free choices \top and q free choices \perp . We can see that $T(p, q) \geq T(p + 1, q) + T(p, q + 1) + 1$ and $T(k, n - 2 - k) \geq 0$. The following solution to this recurrence can be found in [13].

Corollary 1. *There is an $f \in \Omega(n^{k/16})$ s.t. every $\text{p-Res}^*(1)$ refutation of RVIP_n is of size $\geq f(n)$.*

We may increase the number of relativizing predicates to define d -RVIP $_n$.

$$\begin{array}{ll}
R_1^1, \dots, R_1^d, P_{1,1}, R_n^1, \dots, R_n^d \neg P_{n,j} & j \in [n] \\
\bigvee_{l>i, m \in [n]} S_{i,j,l,m} & i, j \in [n] \\
\neg S_{i,j,l,m} \vee \neg R_i^1 \vee \dots \vee \neg R_i^d \vee \neg P_{i,j} \vee R_l^1 & i \in [n-1], j, l, m \in [n] \\
\vdots & \\
\neg S_{i,j,l,m} \vee \neg R_i^1 \vee \dots \vee \neg R_i^d \vee \neg P_{i,j} \vee R_l^r & i \in [n-1], j, l, m \in [n], r \in [d] \\
\neg S_{i,j,l,m} \vee \neg R_i^1 \vee \dots \vee \neg R_i^d \vee \neg P_{i,j} \vee P_{l,m} & i \in [n-1], j, l, m \in [n]
\end{array}$$

We sketch how to adapt the previous argument in order to demonstrate the following.

Corollary 2. *There is an $f \in \Omega(n^{k/16d})$ s.t. every $\text{p-Res}^*(d)$ refutation of RVIP_n^d is of size $\geq f(n)$.*

We use essentially the same Adversary strategy in a branching d -program. We answer questions $\ell_1 \vee \dots \vee \ell_d$ as either forced or free exactly according to the disjunction of how we would have answered the corresponding ℓ_i s, $i \in [d]$, before. That is, if one ℓ_i would give Prover a free choice, then the whole disjunction is given as a free choice. The key point is that once some disjunction involving some subset of R_i^1, \dots, R_i^d or $P_{i,j}$ (never all of these together, of course), is questioned then, on a positive answer to this, the remaining unquestioned variables of this form should be set to \perp . This latter rule introduces the factor of d in the exponent of $n^{k/16d}$.

6.2. Upper bound: a $\text{Res}^*(d+1)$ refutation of RVIP_n^d

We encourage the reader to have a brief look at the simpler, but very similar, refutation of RIP_n in $\text{Res}^*(2)$, of size $O(n^2)$, as depicted in Figure 1.

Proposition 7. *There is a refutation of d -RVIP $_n$ in $\text{Res}^*(d+1)$, of size $O(n^{d+4})$.*

PROOF. We give the branching program for $d := 1$ in Figure 2. The generalization to higher d is clear: substitute questions of the form $\neg R_i \vee \neg P_{i,j}$ by questions of the $\neg R_i^1 \vee \dots \vee \neg R_i^d \vee \neg P_{i,j}$.

7. Final remarks

It is most natural when looking for separators of $\text{p-Res}^*(1)$ and $\text{p-Res}^*(2)$ to look for CNFs, like RVIP_n that we have given. $\text{p-Res}^*(2)$ is naturally able to process 2-DNFs and we may consider $\text{p-Res}^*(1)$ acting on 2-DNFs, when we think of it using any of the clauses obtained from those 2-DNFs by distributivity. In this manner, we offer the following principle as being fpt -bounded for $\text{p-Res}^*(2)$ but not fpt -bounded for $\text{p-Res}^*(1)$. Consider the two axioms $\forall x (\exists y \neg S(x, y) \wedge T(x, y)) \vee P(x)$ and $\forall x, y T(x, y) \rightarrow S(x, y)$. This generates the following system Σ_{PST} of 2-DNFs.

$$\begin{array}{ll}
P_i \vee \bigvee_{j \in [n]} (\neg S_{i,j} \wedge T_{i,j}) & i \in [n] \\
\neg T_{i,j} \vee S_{i,j} & i, j \in [n]
\end{array}$$

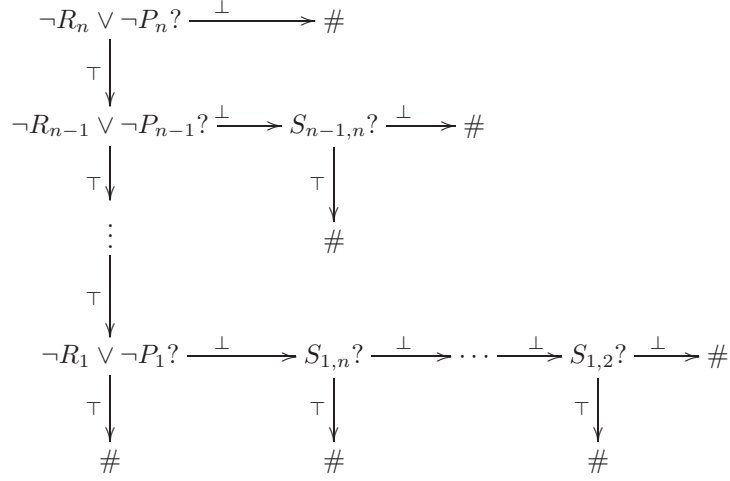


Figure 1: Refutation of RIP_n in $\text{Res}^*(2)$.

Note that the expansion of Σ_{PST} to CNF makes it exponentially larger. It is not hard to see that Σ_{PST} has refutations in $\text{p-Res}^*(2)$ of size $O(kn)$, while any refutation in $\text{p-Res}^*(1)$ will be of size $\geq n^{k/2}$.

All of our upper bounds, i.e. for both RVIP_n and RLNP_n , are in fact polynomial, and do not depend on k . That is, they are fpt -bounded in a trivial sense. If we want examples that depend also on k then we may enforce this easily enough, as follows. For a set of clauses Σ , build a set of clauses Σ'_k with new propositional variables A and $B_1, B'_1, \dots, B_{k+1}, B'_{k+1}$. From each clause $C \in \Sigma$, generate the clause $A \vee C$ in Σ'_k . Finally, augment Σ'_k with the following clauses: $\neg A \vee B_1 \vee B'_1, \dots, \neg A \vee B_{k+1} \vee B'_{k+1}$. If Σ admits refutation of size $\Theta(n^c)$ in $\text{p-Res}^*(d)$ then (Σ'_k, k) admits refutation of size $\Theta(n^c + 2^{k+1})$. The parameterized contradictions so obtained are no longer “strong”, but we could even enforce this by augmenting instead a Pigeonhole principle from $k+1$ to k .

It seems hard to prove $\text{p-Res}(1)$ lower bounds for parameterized k -clique on a random graph [4], but we now introduce a contradiction that looks similar but for which lower bounds should be easier. It is a variant of the Pigeonhole principle which could give us another very natural separation of $\text{p-Res}(1)$ from $\text{p-Res}(2)$. Define the contradiction $\text{PHP}_{k+1,n,k}$, on variables $P_{i,j}$ ($i \in [k+1]$ and $j \in [n]$) and $Q_{i,j}$ ($i \in [n]$ and $j \in [k]$), and with clauses:

$$\begin{array}{ll}
 \neg P_{i,j} \vee \neg P_{l,j} & i \neq l \in [k+1]; j \in [n] \\
 \neg Q_{i,j} \vee \neg Q_{l,j} & i \neq l \in [n]; j \in [k] \\
 \bigvee_{j \in [n]} P_{i,j} & i \in [k] \\
 \neg P_{i,j} \vee \bigvee_{l \in [k]} Q_{j,l} & j \in [n]
 \end{array}$$

We conjecture that this principle, which admits fpt -bounded refutation in $\text{p-Res}(2)$, does not in $\text{p-Res}(1)$.

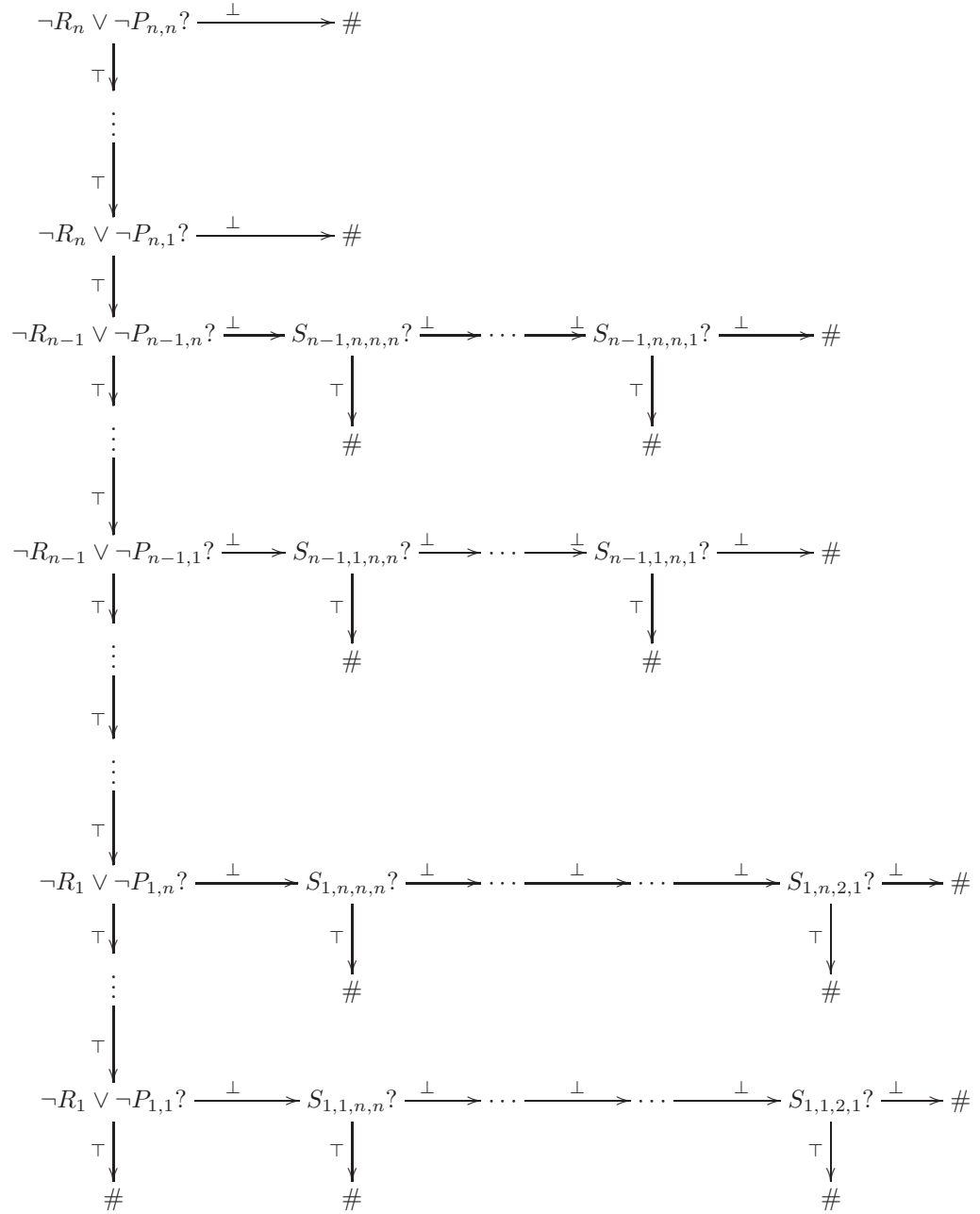


Figure 2: Refutation of RVIP_n in $\text{Res}^*(2)$.

We have left open the technical question as to whether suitably defined, further-relativized versions of $RLNP_n$ can separate $p\text{-Res}(d)$ from $p\text{-Res}(d+1)$. We conjecture that they can.

Finally, it is possible that the results of Section 6 might be derived in a simpler manner using the asymmetric Prover-Delayer game of [5].

Acknowledgements

The authors would like to thank Jan Krajíček for asking the question which led to writing this paper. Many thanks to Søren Riis for the helpful discussions, the first author has had with him. Finally, many thanks to our anonymous referees for pointing out important errors.

References

- [1] A. Atserias and M. Bonet. On the automatizability of resolution and related propositional proof systems. In *16th Annual Conference of the European Association for Computer Science Logic*, 2002.
- [2] Albert Atserias, Moritz Müller, and Sergi Oliva. Lower bounds for dnf-refutations of a relativized weak pigeonhole principle. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:116, 2013.
- [3] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. Hardness of parameterized resolution. Technical report, ECCC, 2010.
- [4] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. Parameterized complexity of DPLL search procedures. In *Theory and Applications of Satisfiability Testing - SAT 2011 - 14th International Conference, SAT 2011*, pages 5–18, 2011.
- [5] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A characterization of tree-like resolution size. *Inf. Process. Lett.*, 113(18):666–671, 2013.
- [6] Olaf Beyersdorff, Nicola Galesi, Massimo Lauria, and Alexander A. Razborov. Parameterized bounded-depth frege is not optimal. *TOCT*, 4(3):7, 2012. Extended abstract appeared in ICALP 2011.
- [7] M. Bonet and N. Galesi. A study of proof search algorithms for resolution and polynomial calculus. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*. IEEE, 1999.
- [8] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979.
- [9] S. Dantchev. Relativisation provides natural separations for resolution-based proof systems. In *Computer Science - Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8-12, 2006, Proceedings*, volume 3967 of *Lecture Notes in Computer Science*, pages 147–158. Springer, 2006.

- [10] S. Dantchev and B. Martin. Parameterized resolution with bounded conjunction. In *Computer Science - Theory and Applications, 8th International Computer Science Symposium in Russia, CSR 2006, Ekaterinburg, Russia*, Lecture Notes in Computer Science. Springer, 2013.
- [11] S. Dantchev and S. Riis. On relativisation and complexity gap for resolution-based proof systems. In *The 17th Annual Conference of the EACSL, Computer Science Logic*, volume 2803 of *LNCS*, pages 142–154. Springer, August 2003.
- [12] Stefan Dantchev, Barnaby Martin, and Stefan Szeider. Parameterized proof complexity. In *48th IEEE Symp. on Foundations of Computer Science*, pages 150–160, 2007.
- [13] Stefan Dantchev, Barnaby Martin, and Stefan Szeider. Parameterized proof complexity. *Computational Complexity*, 20, 2011.
- [14] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, 1999.
- [15] J.L. Esteban, N. Galesi, and J. Mesner. On the complexity of resolution with bounded conjunctions. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, 2002.
- [16] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer Verlag, 2006.
- [17] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.
- [18] J. Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Cambridge University Press, 1995.
- [19] J. Krajicek. Combinatorics of first order structures and propositional proof systems. *Archive for Mathematical Logic*, 43(4), 2004.
- [20] Barnaby Martin. Parameterized proof complexity and W[1]. CoRR: arxiv.org/abs/1203.5323, 2012.
- [21] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, New York, NY, USA, 1995.
- [22] P. Pudlák. Proofs as games. *American Mathematical Monthly*, pages 541–550, June-July 2000.
- [23] S. Riis. A complexity gap for tree-resolution. *Computational Complexity*, 10:179–209, 2001.

- [24] S.M. Riis and M. Sitharam. Generating hard tautologies using predicate logic and the symmetric group. *Logic Journal of the IGPL*, 8(6):787–795, 2000.
- [25] Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-dnf resolution. *SIAM J. Comput.*, 33(5):1171–1200, 2004.